

Sponsors and Partners

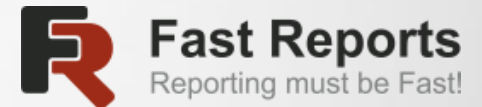
Strategic Sponsors

The logo for adform, featuring the word "adform" in a lowercase, sans-serif font. The letter "o" is stylized with a blue and yellow circular graphic element.

Gold Sponsors



Silver Sponsors





Dependency Injection in .Net : Revisit

Waldis Iljuczonok



Wałdis Ijuczonok

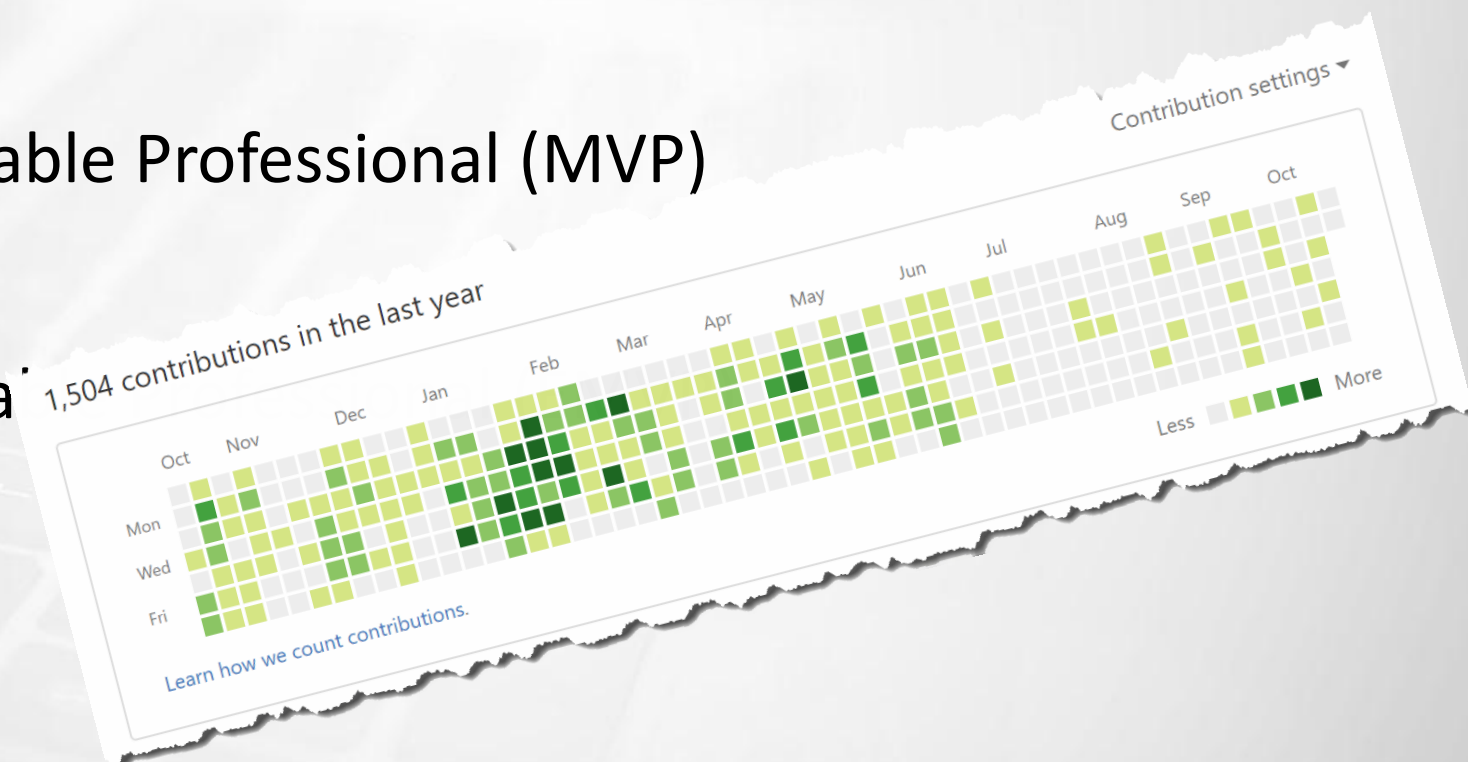
Software Architect (@tech_fellow)



Microsoft Most Valuable Professional (MVP)



EpiServer Most Valuable Professional (EMVP)



The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'wp_get_attachment_url' and 'wp_get_attachment_image'. The keyboard is visible in the lower half of the image, with keys like 'R', 'E', 'T', 'G', 'H', 'Y', 'U', 'I', 'O', 'P' and numbers '4', '5', '6', '7', '8' visible. The overall image is out of focus, with the word 'Why?' centered in the foreground.

Why?



WORKING WITH LEGACY CODE

Plan

- Patterns
- Anti-Patterns
- Applying DI
- Personal Experience with DI

Definition

The background of the image is a blurred photograph of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'first', 'last', and 'wp_get_attachment_url'. The keyboard is visible in the lower half of the image, with keys like '5', 'T', 'G', '8', 'R', 'E', 'V', 'N' being discernible. The overall image has a soft, out-of-focus aesthetic.

What is dependency?

“

DI is a set of principles and patterns
that enable loose coupling



```
for ($loop = 0; $loop < $columns; $loop++)  
{  
    $columns = 'first';  
    $columns = 'last';  
    $image_link = wp_get_attachment_url($attachment_id);  
    $image_link = wp_get_attachment_url($attachment_id);  
    $image_class = esc_attr(implode(' ', $image_class));  
    $image_title = esc_attr(implode(' ', $image_title));  
    $image_html = sprintf('div class="%s" style="width: %spx; height: %spx; text-align: center; vertical-align: middle;">  
      
    </div>', $image_class, $image_width, $image_height, $image_title, $image_link);  
    $loop++;  
}
```

Object Composition Inversion of Control (IoC)

```
for ( $loop = 0 || $loop = $columns
  $images = 'first';
  (( ($loop = 1) % $columns == 0 )
  $images = 'last';
  $image_link = wp_get_attachment_url( $attachment_id );
  if ( ! $image_link )
    continue;
  $image = wp_get_attachment_image( $attachment_id, $size );
  $image_class = esc_attr( $image );
  $image_title = esc_attr( $image );
  printf( 'div class="slide %s"',
    $image_class, $image_title );
  printf( 'div class="slide %s"',
    $image_class, $image_title );
  $loop++;
}
```

S.O.L.I.D.

```
if ( $loop == 0 || $loop == $columns )  
    $images[] = 'first';  
if ( ( $loop == 1 ) % $columns == 0 )  
    $images[] = 'last';  
$image_link = wp_get_attachment_url( $attachment_id );  
if ( ! $image_link )  
    continue;  
$image = wp_get_attachment_image_src( $attachment_id, $size );  
$image_class = esc_attr( $image );  
$image_title = esc_attr( $image );  
$image_html = sprintf( 'div class="%s" style="display: inline-block; vertical-align: top; margin-right: 10px; margin-bottom: 10px;">  
      
    </div>';  
$loop++;  
}
```


Patterns

Constructor Injection




Constructor Injection

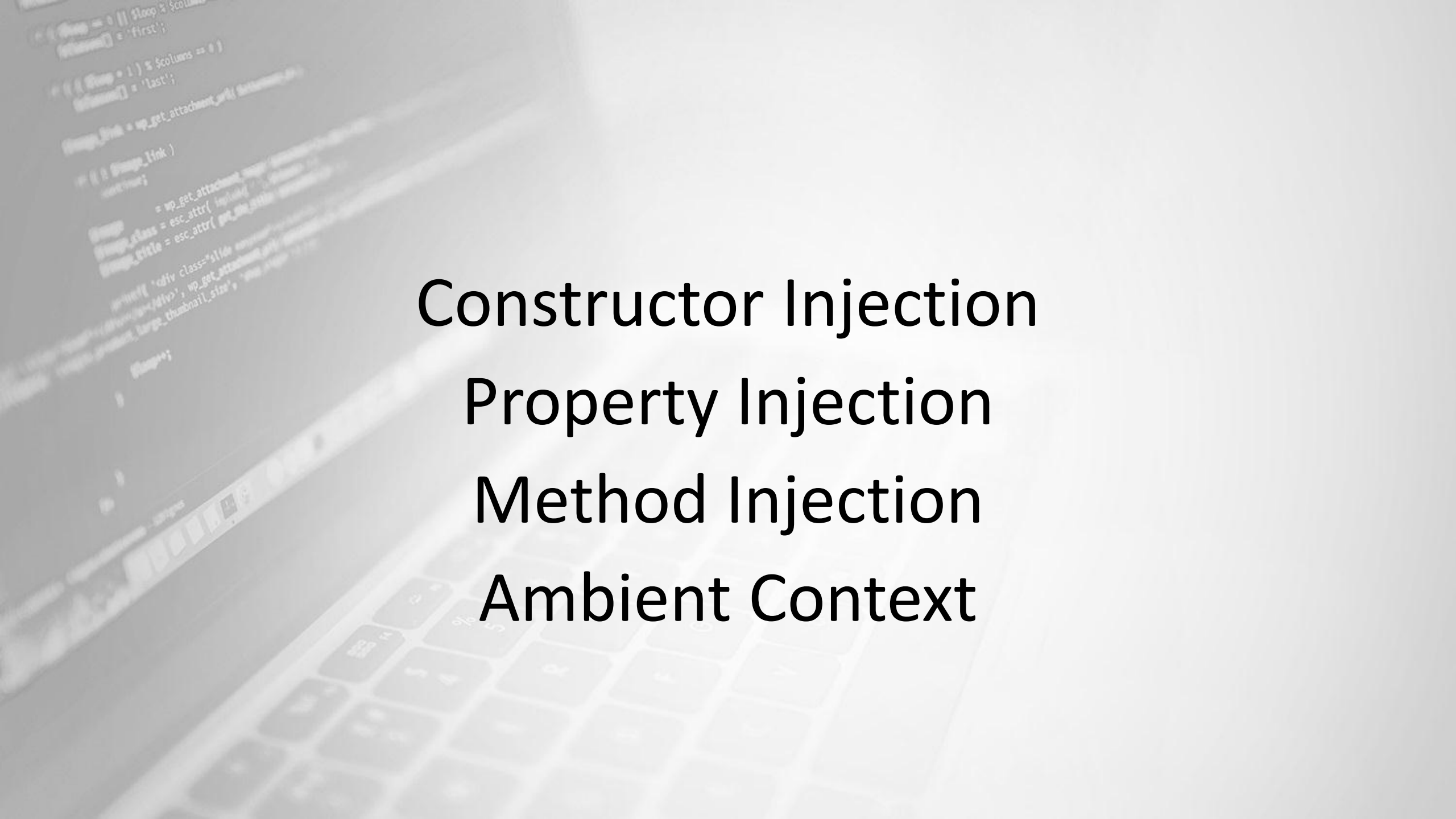
Property Injection




Constructor Injection
Property Injection
Method Injection



Constructor Injection
Property Injection
Method Injection
Ambient Context

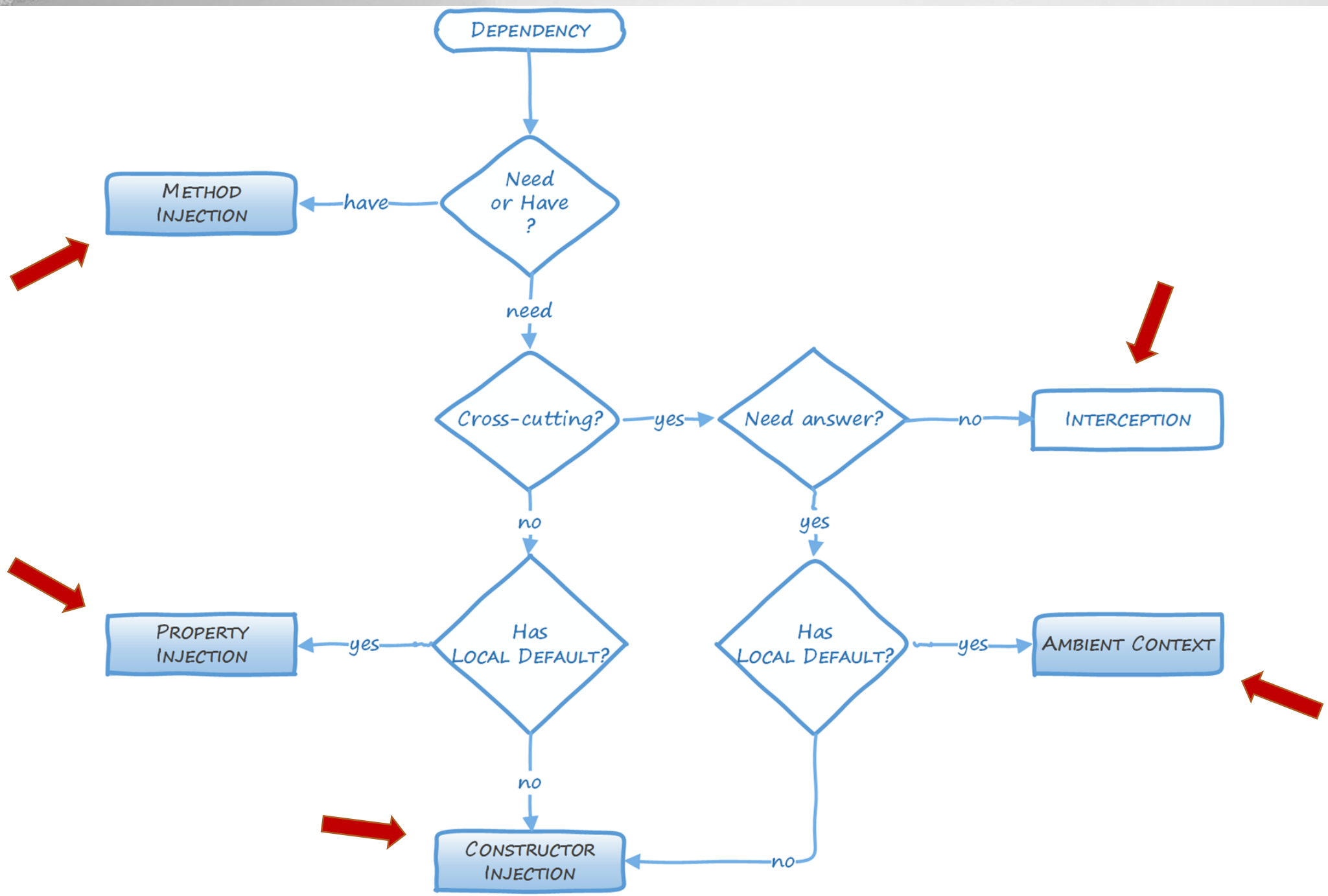


```
for ( $loop = 0 || $loop - $columns )  
    {  
        $columns = 'first';  
        {  
            $loop = 1 } % $columns == 0 )  
            $columns = 'last';  
        $image_link = wp_get_attachment_url( $attachment_id );  
        {  
            $image_link )  
            $image = wp_get_attachment_image( $attachment_id, $image_size );  
            $image_class = esc_attr( $image );  
            $image_title = esc_attr( $image );  
            $image_html = sprintf( '



 %s', $image_class, $image_title, $image_link );  
            $output .= $image_html . $text . "\n";  
        }  
        $loop++;  
    }  
}
```

Constructor Injection
Property Injection
Method Injection
Ambient Context



The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'wp_get_attachment_url' and 'wp_get_attachment_image'. The keyboard is visible in the foreground, with keys like 'T', 'G', '8', 'R', 'F', 'V', '5', and '4' being discernible. The overall image is out of focus, creating a soft, artistic effect.

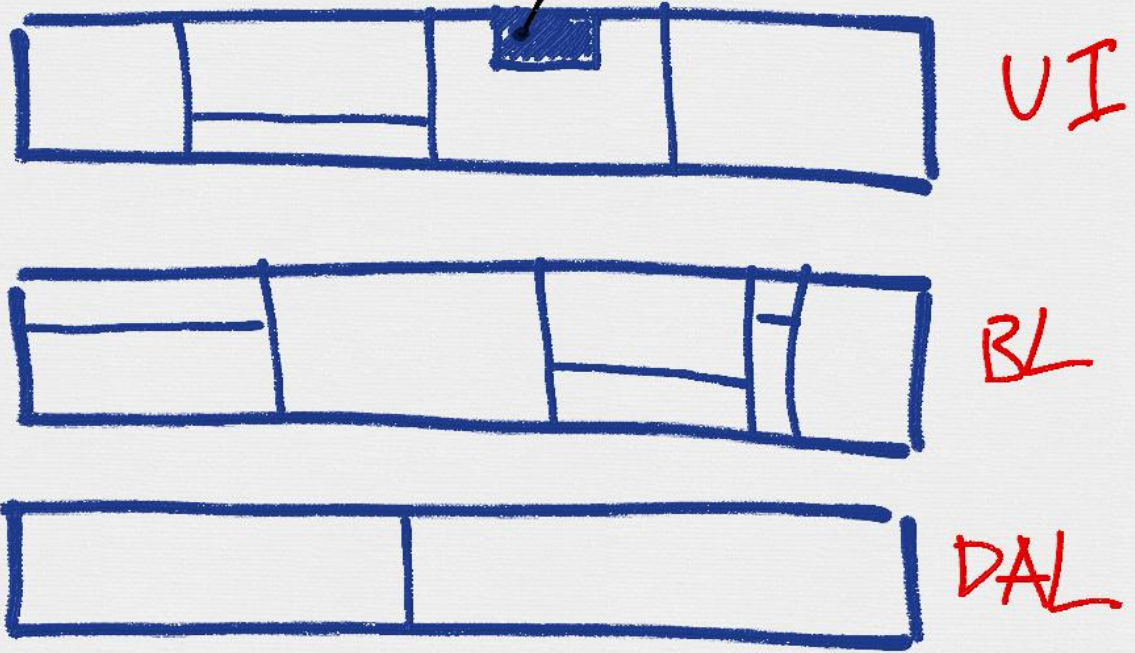
Anti-Patterns

Control Freak

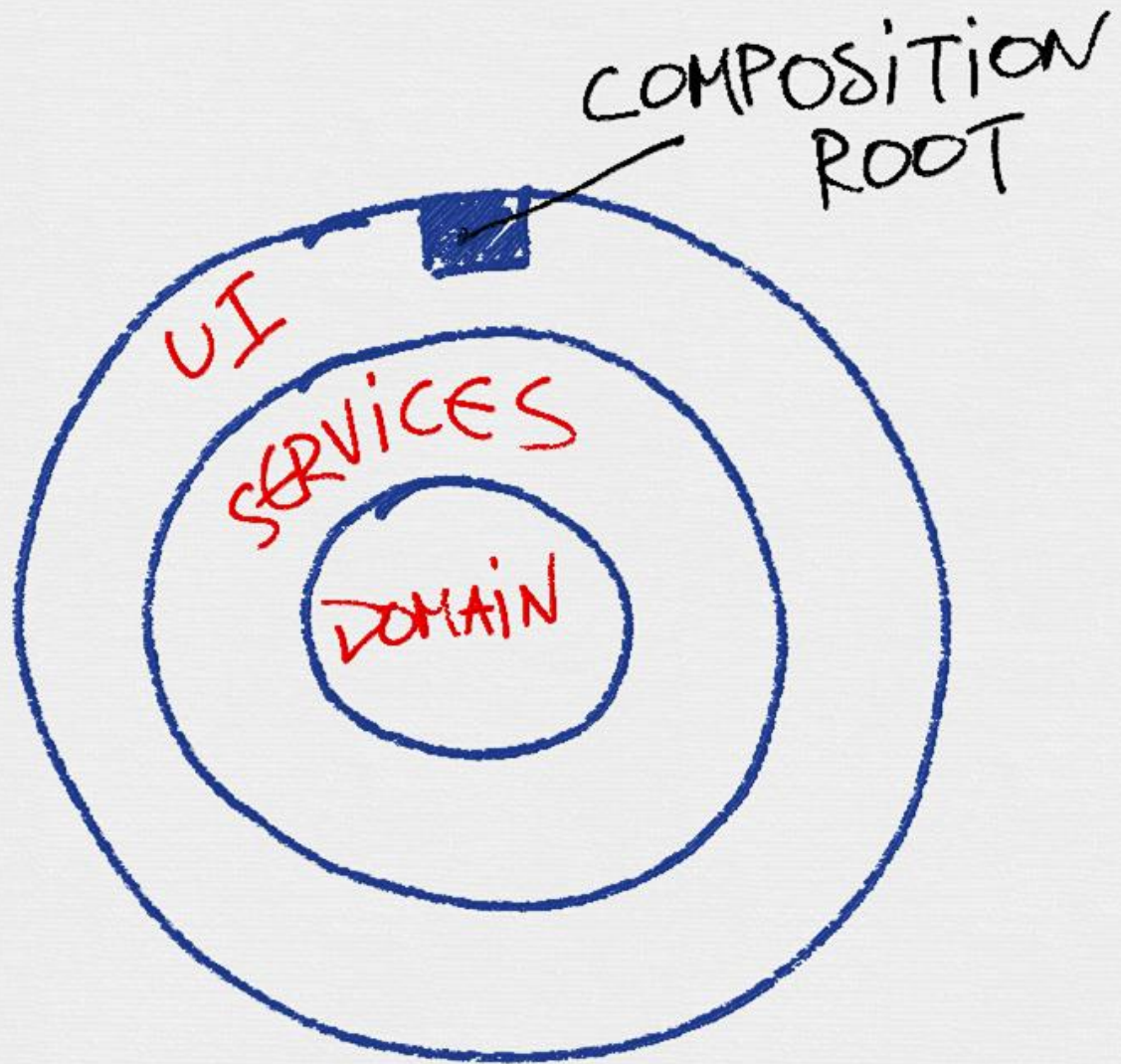


Composition Root

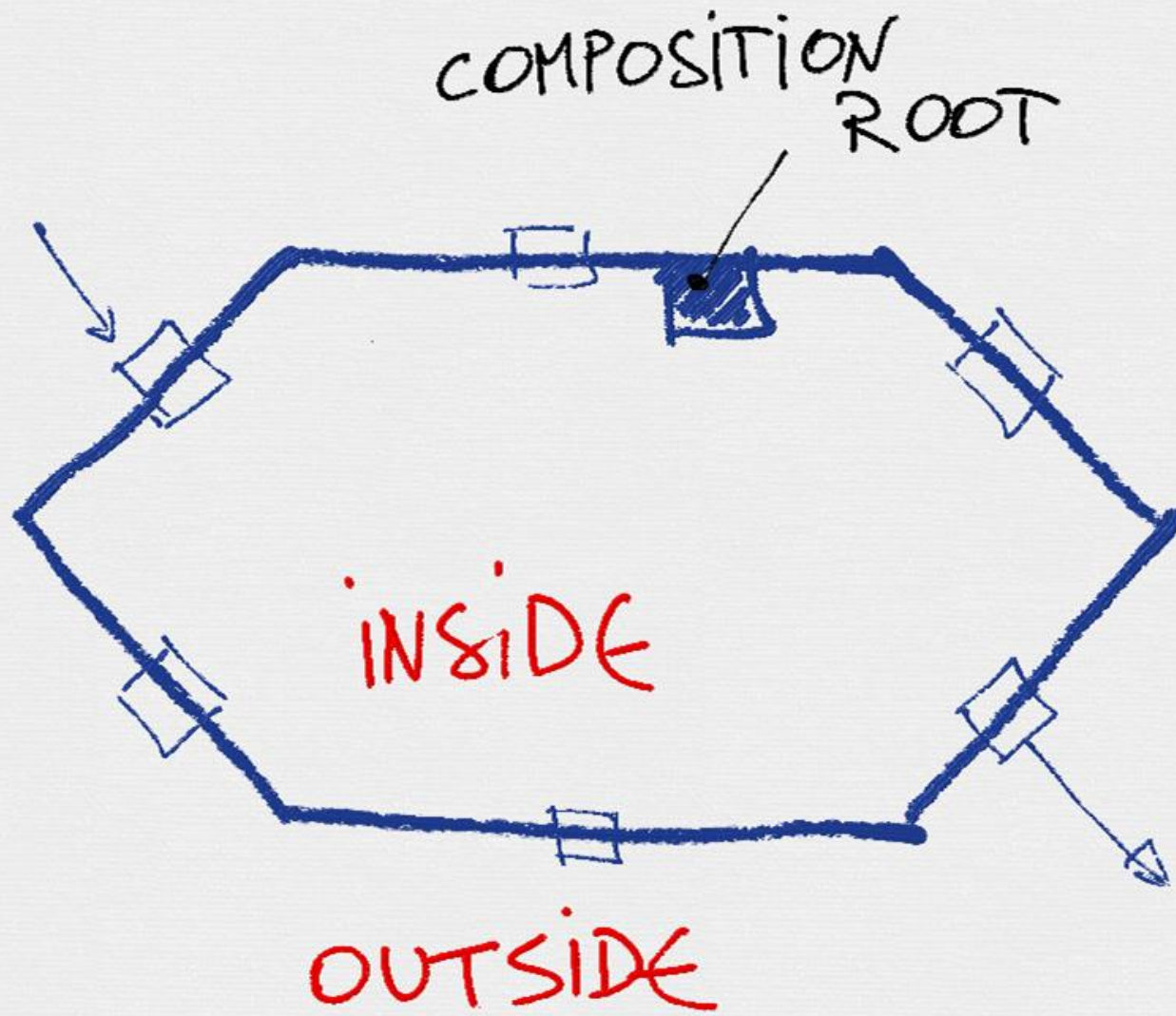
COMPOSITION ROOT



Layered



Onion



hexagonal


```
if ( $loop == 0 || $loop == $columns )  
  $item = 'first';  
if ( ( $loop == 1 ) % $columns == 0 )  
  $item = 'last';  
$img_link = wp_get_attachment_url( $attachment_id );  
if ( ! $img_link )  
  continue;  
$img = wp_get_attachment_image( $attachment_id, $size );  
$img_class = esc_attr( $img );  
$img_title = esc_attr( $img );  
$html .= <div class="slide"></div>;  
wp_get_attachment_image( $attachment_id, $size );  
$html .= <div class="slide"></div>;
```

Global.asax & IControllerFactory

Startup.cs & RouterMiddleware

Application.OnStartup

..rely on framework..




```
for ($loop = 0; $loop < $columns; $loop++) {  
    $columns = 'first';  
    // ...  
    $columns = 'last';  
    $image_link = wp_get_attachment_url($attachment_id);  
    // ...  
    $image = wp_get_attachment_image($attachment_id, $size);  
    $image_class = esc_attr( $image_class );  
    $image_title = esc_attr( $image_title );  
    // ...  
    $loop++;  
}
```

Control Freak Bastard Injections



Control Freak
Bastard Injections
Constrained Construction



```
for ($loop = 0; $loop < $columns; $loop++) {  
    $columns = 'first';  
    // ...  
    $columns = 'last';  
    $image_link = wp_get_attachment_url($attachment_id);  
    // ...  
    $image = wp_get_attachment_image($attachment_id, $size);  
    $image_class = esc_attr(implode(' ', $classes));  
    $image_title = esc_attr(get_the_title($attachment_id));  
    // ...  
    $loop++;  
}
```

Control Freak
Bastard Injections
Constrained Construction
Service Locator

Don't ask, we will call you!

“Hollywood Principle”

CommonServiceLocator - x

https://commonservicelocator.codeplex.com/releases/view/118130

CodePlex Project Hosting for Open Source Software Register Sign In Search all projects

Microsoft® patterns & practices proven practices for predictable results

CommonServiceLocator


HOME SOURCE CODE **DOWNLOADS** DOCUMENTATION DISCUSSIONS ISSUES PEOPLE LICENSE

Subscribe

Common Service Locator 1.2

Rating: ★★★★★ Based on 3 ratings	Released: Feb 5, 2014
Reviewed: 2 reviews	Updated: Feb 6, 2014 by fcheung
Downloads: 7499	Dev status: Stable ?

RECOMMENDED DOWNLOAD

 [Microsoft.Practices.ServiceLocation.zip](#)
application, 16K, uploaded Feb 5, 2014 - 7499 downloads

RELEASE NOTES

This release of the Common Service Locator includes a portable class library that targets the following platforms:

- » .NET 4 and higher
- » Windows Phone 8
- » Windows Store apps (Windows 8) and higher
- » Silverlight 5


OTHER DOWNLOADS

Released | Planned

- ★ **Common Service Locator 1.2**
Feb 5, 2014, Stable
★★★★★
- [Common Service Locator 1.0](#)
Sep 30, 2008, Stable
★★★★★

Release notifications

[Sign in](#) to display notification settings.



```
for ($loop = 0; $loop < $columns; $loop++) {  
    $columns = 'first';  
    // ...  
    $columns = 'last';  
    $image_link = wp_get_attachment_url($attachment_id);  
    // ...  
    $image = wp_get_attachment_image($attachment_id, $size);  
    $image_class = esc_attr(implode(' ', $classes));  
    $image_title = esc_attr(get_the_title($attachment_id));  
    // ...  
    $loop++;  
}
```

Control Freak

Bastard Injections

Constrained Construction

Service Locator

The background is a blurred image of a laptop. The top-left corner shows a code editor window with PHP code. The code includes a loop structure with comments like 'first' and 'last', and function calls such as 'wp_get_attachment_url'. The rest of the image shows the laptop's keyboard and trackpad, which are out of focus.

Applying DI



Pure DI (aka Poor Man's DI)

DI (IoC) Container

Autofac
Caliburn.Micro
Catel
DryIoc
DryIocZero
Dynamo
fFastInjector
Funq
Grace
Griffin
HaveBox
IfInjector
LightCore
LightInject
LinFu
Maestro
Mef
Mef2
MicroSliver
Microsoft.Extensions.DependencyInjection
Mugen
Munq
Ninject
Petite
SimpleInjector
Spring.NET
Stashbox
StructureMap
StyleMvvm
TinyIoC
Unity
Windsor

The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'wp_get_attachment_url' and 'wp_get_attachment_image'. The keyboard is visible in the lower half of the image, with keys like 'R', 'E', 'F', 'V', 'G', 'N', 'B', 'H', '5', '4', '3', '2', '1', '0', and '9' being discernible. The overall image has a soft, out-of-focus aesthetic.

Register-Resolve-Release

Dependency Lifetime

(captive dependencies)

The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'first', 'last', and 'wp_get_attachment_url'. The keyboard is visible in the lower half of the image, with keys like 'T', 'G', '8', 'R', 'F', 'V', 'S', and '4' being discernible. The overall image is in grayscale and has a soft, out-of-focus appearance.

Personal Experience with DI

The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'first', 'last', and 'wp_get_attachment_url'. The keyboard is visible in the lower half of the image, with keys like '5', 'T', 'G', '8', 'R', 'E', 'V' clearly visible. The overall image is in grayscale and has a soft, out-of-focus effect.

Conforming Container

The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'wp_get_attachment_url' and 'wp_get_attachment_image'. The keyboard is visible in the foreground, with keys like 'T', 'G', '8', 'N', 'R', 'F', 'V', '5', and '4' being discernible. The overall image is out of focus, creating a soft, bokeh effect.

FeatureSwitch

GitHub repository header for **valdisiljuconoks / FeatureSwitch**. It includes navigation links for Pull requests, Issues, and Gist, along with repository statistics: 3 Unwatch, 27 Stars, and 3 Forks. Below the header are tabs for Code, Issues (7), Pull requests (0), Wiki, Pulse, Graphs, and Settings.

FeatureSwitch is library that should reduce amount of time and code required to implement feature switching in your projects. — Edit

Repository statistics: 95 commits, 2 branches, 3 releases, 2 contributors.

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

Commit	Description	Time
valdisiljuconoks	Added non-generic overloads for scripts and styles renderer. This clo...	Latest commit 20de419 on Oct 21, 2015
.nuget	Removed StructureMap dependnecy. This closes #4	2 years ago
.vs/config	Added some unit test for new functionality	11 months ago
FeatureSwitch.AspNet.Mvc	Added local and remote request features. This closes #11	2 years ago
FeatureSwitch.AspNet.Mvc5	Added role based strategy. This closes #18.	11 months ago
FeatureSwitch.AspNet.Sample	Upgrade EPiServer to 9.x	11 months ago
FeatureSwitch.Azure	Added Azure support for cloud configuration	11 months ago
FeatureSwitch.EPiServer.Cms80	Few bug fixes	10 months ago
FeatureSwitch.Glimpse	Removed content script	2 years ago
FeatureSwitch.Ninject.Tests	Added empty constructor	2 years ago
FeatureSwitch.Ninject	Fixed some namings, references and copyright	2 years ago
FeatureSwitch.StructureMap.2.6	Updated packages to new version. This closes #6.	2 years ago
FeatureSwitch.StructureMap.Tests	Upgrade EPiServer to 9.x	11 months ago
FeatureSwitch.StructureMap	Upgrade EPiServer to 9.x	11 months ago
FeatureSwitch.Unity.Tests	Updated packages to new version. This closes #6.	2 years ago
FeatureSwitch.Unity	Updated packages to new version. This closes #6.	2 years ago
FeatureSwitch.Web.Optimization	Added non-generic overloads for scripts and styles renderer. This clo...	10 months ago
FeatureSwitch	Added some unit test for new functionality	11 months ago
FeatureSwitchTests	Added some unit test for new functionality	11 months ago

- * .Ninject
- * .StructureMap
- * .Unity
- * .<your-container>

Repository	Commit Message	Time
FeatureSwitch.AspNet.Mvc5	Added role based strategy. T	2 years ago
FeatureSwitch.AspNet.Sample	Upgrade EPiServer to 9.x	2 years ago
FeatureSwitch.Azure	Added Azure support for clo	2 years ago
FeatureSwitch.EPiServer.Cms80	Few bug fixes	2 years ago
FeatureSwitch.Glimpse	Removed content script	2 years ago
FeatureSwitch.Ninject.Tests	Added empty constructor	2 years ago
FeatureSwitch.Ninject	Fixed some namings, referen	2 years ago
FeatureSwitch.StructureMap.2.6	Updated packages to new ve	2 years ago
FeatureSwitch.StructureMap.Tests	Upgrade EPiServer to 9.x	2 years ago
FeatureSwitch.StructureMap	Upgrade EPiServer to 9.x	2 years ago
FeatureSwitch.Unity.Tests	Updated packages to new ve	2 years ago
FeatureSwitch.Unity	Updated packages to new ve	2 years ago



- * .Ninject
- * .StructureMap
- * .Unity
- * .<your-container>




Constructor Over-Injection

The background is a blurred image of a laptop. The screen on the left shows lines of code, likely PHP, with some text like 'wp_get_attachment_url' and 'wp_get_attachment_image'. The keyboard is visible in the foreground, with keys like '5', 'T', 'G', '8', 'R', 'E', 'V', 'N' being discernible. The overall image is in grayscale and has a soft, out-of-focus effect.

Role Interfaces & Service Locator



AbstractFactory



Abstract Factory vs Service Locator


```
public interface IServiceLocator
{
    T Resolve<T>(object context);
}
```

ServiceLocator

```
public interface IAbstractFactory<T>
{
    T Resolve(object context);
}
```

AbstractFactory

```
public interface IServiceLocator
{
    T Resolve<T>(object context);
}
```

ServiceLocator

```
public interface IAbstractFactory<T>
{
    T Resolve(object context);
}
```

AbstractFactory

The background is a blurred image of a laptop. The screen on the left shows PHP code with comments and function calls like `wp_get_attachment_url()` and `esc_attr()`. The keyboard is visible in the lower half of the image, with keys like '5', 'T', 'G', and '8' being discernible. The overall image is in grayscale and has a soft, out-of-focus appearance.

DI aware [Attributes]



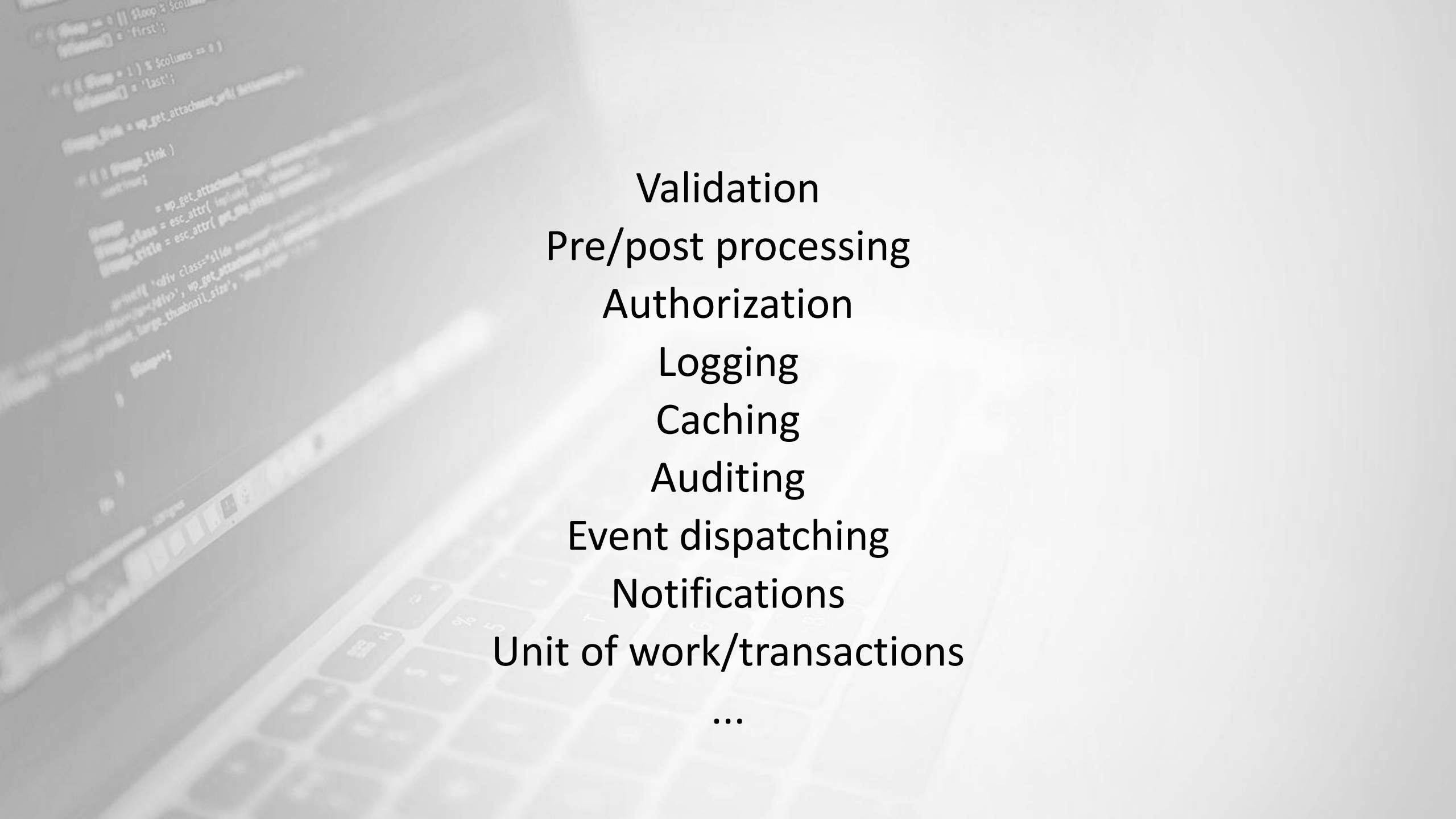
Extra Bonus with DI



Aspects



russian doll (“матрёшка”)
model



Validation
Pre/post processing
Authorization
Logging
Caching
Auditing
Event dispatching
Notifications
Unit of work/transactions

...




The background features a blurred image of a laptop. The top-left corner shows a code editor with PHP code, including a loop and function calls like `wp_get_attachment_url()`. The bottom-left shows a portion of a keyboard with keys like '5', 'T', 'G', and '8' visible.

Decorators

vs

Interceptors

```
for ( $loop = 0 || $loop = $columns  
  $columns ) = 'first';  
for ( ( $loop = 1 ) % $columns = 0 )  
  $columns ) = 'last';  
$image_link = wp_get_attachment_url( $attachment_id );  
if ( ! $image_link )  
  continue;  
$image = wp_get_attachment_image( $attachment_id, $size );  
$image_class = esc_attr( 'image' );  
$image_title = esc_attr( get_the_title( $attachment_id ) );  
$image_html = sprintf( '



', $image_class, $image_title, $image );  
$loop++;
```

Create loose-coupling code
Demand dependencies from consumer
Leverage DI container features
Look for anti-patterns

```
for ( $loop = 0 || $loop = $columns; $loop < $columns; $loop++)
{
    $columns = 'first';
    $columns = 'last';
    $image_link = wp_get_attachment_url( $attachment_id );
    if ( ! $image_link )
    {
        $image = wp_get_attachment_image( $attachment_id, $size );
        $image_class = esc_attr( $image );
        $image_title = esc_attr( $image );
        $image_html = sprintf( 'div class="%s" style="width: %spx; height: %spx; text-align: center; vertical-align: middle;">', $image_class, $image_width, $image_height );
        $image_html .= sprintf( 'img alt="%s" data-attachment-id="%s" data-permalink="%s" data-size="%s" data-full="%s" data-orig-file="%s" data-orig-size="%s" data-cropped="%s" data-comments="%s" data-retina="%s"/>', $image_title, $attachment_id, $image_link, $image_size, $image_full, $image_file, $image_width, $image_height, $image_cropped, $image_comments, $image_retina );
    }
    $loop++;
}
```

Be a BOY SCOUT!



THANK YOU!



.NET **DEVELOPER** **DAYS**

net.developerdays.pl
[@DeveloperDaysPL](https://twitter.com/DeveloperDaysPL)

Sponsors and Partners

Strategic Sponsors

The logo for adform, featuring the word "adform" in a grey sans-serif font. The letter "o" is stylized as a circle with a yellow-to-blue gradient.

Gold Sponsors



Silver Sponsors

